



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/870,624	05/31/2001	Scott J. Broussard	AUS920010268US1	1775
35617	7590	01/29/2004	EXAMINER	
CONLEY ROSE, P.C. P.O. BOX 684908 AUSTIN, TX 78768			BONSHOCK, DENNIS G	
			ART UNIT	PAPER NUMBER
			2173	3
DATE MAILED: 01/29/2004				

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/870,624

Applicant(s)

BROUSSARD, SCOTT J.

Examiner

Dennis G Bonshock

Art Unit

2173

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 31 May 2001.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-20 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-20 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. §§ 119 and 120

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
* See the attached detailed Office action for a list of the certified copies not received.
- 13) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application) since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.
a) ☐ The translation of the foreign language provisional application has been received.
- 14) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121 since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s) 2.
- 4) ☐ Interview Summary (PTO-413) Paper No(s). _____
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: _____

DETAILED ACTION

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

2. Claims 1-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lewallen, Patent # 6,675,230, and Fowler, "Mixing heavy and light components."

3. With regard to claim 1, Lewallen teaches a display (see column 15, lines 47-53), a graphical user interface (see column 1, lines 38-41), a processor for implementing the embodiments of the invention (see claim 25), the system being implemented on a windows based operating system (see column 1, lines 40-45), implementing two different APIs to generate objects in the same application program (see column 3, lines 25-40), and the images being substantially identical (see column 6, lines 50-67). Lewallen also teaches the use of two particular Java APIs (AWT and Swing), but doesn't get in to the specifics of each of the APIs, in which one is generated at a first time, being independent of the operating system; and one, generated at a second time, being dependent on the operating system. Fowler teaches a system of mixing two APIs similar to that of Lewallen (see page 1, paragraphs 1 and 2), but further teaches specifics of AWT and Swing, where AWT uses heavyweight components (components that associate with native screen resources, and thus are dependent on the operating

Art Unit: 2173

system), and where Swing uses lightweight components (components that borrow from screen resources of an ancestor, and thus are independent of the operating system) (see page 2, paragraphs 1 and 3). Fowler further teaches, on page 7, the components having the same look and feel. It would have been obvious to one of ordinary skill in the art, having the teachings of Lewallen and Fowler before him at the time the invention was made to modify the system of using two APIs in one application program, of Lewallen, to include the use of the AWT and Swing APIs, as did Fowler. One would have been motivated to make such a combination because AWT and Swing are two well-known APIs that are also mentioned in Lewallen (see column 1, line 39 and column 10, line 40). Also the use of the combination of AWT and Swing, which have been proven to be usable together (see Fowler page 1, paragraph 2), would add an element of platform independence to the invention of Lewallen.

4. With regard to claim 2, which teaches the first and second images having the same look and feel, Lewallen further teaches, in column 6, lines 50-67, the images having a similar look and feel.

5. With regard to claims 3 and 11, which teach the first and second images comprising pixels presented upon the display via the graphical user interface associated with the application program, Lewallen further teaches, in claim 4, a first or second user interface program affecting the display of the user interface.

6. With regard to claims 4 and 12, which teach the object being selected from a group comprising buttons, list boxes, and slide bars on which a pointer device can be

directed by a user, Fowler further teaches, on page 1, paragraph 1, Swing an AWT containing components such as buttons, lists, and the like.

7. With regard to claims 5 and 13, which teach an application program written in Java programming language, Lewallen further teaches, in column 3, lines 52-55, preferred embodiments which use cross-platform languages, such as Java.

8. With regard to claims 6 and 14, which teach the software component comprising a java application program interface consisting of an abstract windowing toolkit (AWT) during a second time, Lewallen further teaches, the use of two APIs in a system, with one implemented inside of the other, he also mentions Java's use of AWT and Swing, but does not teach using these two APIs together. Fowler teaches a system in which two APIs can be used in the same application program similar to that of Lewallen, but also teaches the user of Swing and AWT in the same application program (see page 1, paragraph 2). It would have been obvious to one of ordinary skill in the art, having the teachings of Lewallen and Fowler before him at the time the invention was made to modify the system of using two APIs in one application program, of Lewallen, to include the use of the AWT and Swing APIs, as did Fowler. One would have been motivated to make such a combination because AWT and Swing are two well-known APIs that are also mentioned in Lewallen (see column 1, line 39 and column 10, line 40). Also the use of the combination of AWT and Swing, which have been proven to be usable together (see Fowler page 1, paragraph 2), would add an element of platform independence to the invention of Lewallen.

9. With regard to claims 7 and 15, which teach the software component comprising a java application program interface consisting of a Swing application program interface during a first time, Lewallen further teaches, the use of two APIs in a system, with one implemented inside of the other, he also mentions Java's use of AWT and Swing, but does not teach using these two APIs together. Fowler teaches a system in which two APIs can be used in the same application program similar to that of Lewallen, but also teaches the user of Swing and AWT in the same application program (see page 1, paragraph 2). It would have been obvious to one of ordinary skill in the art, having the teachings of Lewallen and Fowler before him at the time the invention was made to modify the system of using two APIs in one application program, of Lewallen, to include the use of the AWT and Swing APIs, as did Fowler. One would have been motivated to make such a combination because AWT and Swing are two well-known APIs that are also mentioned in Lewallen (see column 1, line 39 and column 10, line 40). Also the use of the combination of AWT and Swing, which have been proven to be usable together (see Fowler page 1, paragraph 2), would add an element of platform independence to the invention of Lewallen.

10. With regard to claims 8 and 16, which teach the operating system comprising a Windows, Unix, or OS/2 computer operating system, Lewallen further teaches, in column 1, lines 40-45, the use of operating systems such as Windows, OS/2, etc.

11. With regard to claims 9 and 17, which teach the first and second images presenting the same look and feel upon the display independent of the operating system, Lewallen further teaches, in column 1, lines 33-48, that the java look and feel,

that of AWT and Swing, is significantly different from the look and feel implement in the native operating system, in that Java provides a distinctive platform-independent look and feel.

12. With regard to claim 10, Lewallen teaches a method of displaying (see column 15, lines 47-53), running an application program upon a computer and under an operating system (see column 4, lines 14-17), implementing two different APIs to display two objects in the same application program (see column 3, lines 25-40), and the images having substantially the same look and feel (see column 6, lines 50-67). Lewallen also teaches the use of two particular Java APIs (AWT and Swing), but doesn't get in to the specifics of each of the APIs, in which one is generated at a first time, being dependent on the operating system; and one, generated at a second time, being independent on the operating system. Fowler teaches a system of mixing two APIs similar to that of Lewallen (see page 1, paragraphs 1 and 2), but further teaches specifics of AWT and Swing, where AWT uses heavyweight components (components that associate with native screen resources, and thus are dependent on the operating system), and where Swing uses lightweight components (components that borrow from screen resources of an ancestor, and thus are independent of the operating system) (see page 2, paragraphs 1 and 3). Fowler further teaches, on page 7, the components having the same look and feel. It would have been obvious to one of ordinary skill in the art, having the teachings of Lewallen and Fowler before him at the time the invention was made to modify the system of using two APIs in one application program, of Lewallen, to include the use of the AWT and Swing APIs, as did Fowler. One would

have been motivated to make such a combination because AWT and Swing are two well-known APIs that are also mentioned in Lewallen (see column 1, line 39 and column 10, line 40). Also the use of the combination of AWT and Swing, which have been proven to be usable together (see Fowler page 1, paragraph 2), would add an element of platform independence to the invention of Lewallen.

13. With regard to claim 18, Lewallen teaches a computer-readable storage device, comprising: an operating system (see column 4, lines 14-17), and application program adapted for executing code of a software component (ex: APIs) (see column, 4, lines 14-17), implementing two different APIs to generate objects in the same application program (see column 3, lines 25-40), and the images being substantially identical (see column 6, lines 50-67). Lewallen also teaches the use of two particular Java APIs (AWT and Swing), but doesn't get in to the specifics of each of the APIs, in which one is generated at a first time, being independent of the operating system and one, generated at a second time, being dependent on the operating system. Fowler teaches a system of mixing two APIs similar to that of Lewallen (see page 1, paragraphs 1 and 2), but further teaches specifics of AWT and Swing, where AWT uses heavyweight components (components that associate with native screen resources, and thus are dependent on the operating system), and where Swing uses lightweight components (components that borrow from screen resources of an ancestor, and thus are independent of the operating system) (see page 2, paragraphs 1 and 3). Fowler further teaches, on page 7, the components having the same look and feel. It would have been obvious to one of ordinary skill in the art, having the teachings of Lewallen and Fowler

before him at the time the invention was made that the first image would overwrite an image upon the display previous to the first image and that the first image can't overwrite the second image during the first time (because it hasn't been generated yet) and also to modify the system of using two APIs in one application program, of Lewallen, to include the use of the AWT and Swing APIs, as did Fowler. One would have been motivated to make such a combination because AWT and Swing are two well-known APIs that are also mentioned in Lewallen (see column 1, line 39 and column 10, line 40). Also the use of the combination of AWT and Swing, which have been proven to be usable together (see Fowler page 1, paragraph 2), would add an element of platform independence to the invention of Lewallen.

14. With regard to claim 19, which teaches the software component comprising a java application program interface consisting of a Swing application program interface during a first time, Lewallen further teaches, the use of two APIs in a system, with one implemented inside of the other, he also mentions Java's use of AWT and Swing, but does not teach using these two APIs together. Fowler teaches a system in which two APIs can be used in the same application program similar to that of Lewallen, but also teaches the user of Swing and AWT in the same application program (see page 1, paragraph 2). Fowler further teaches specifics of AWT and Swing, where AWT uses heavyweight components (components that associate with native screen resources, and thus are dependent on the operating system), and where Swing uses lightweight components (components that borrow from screen resources of an ancestor, and thus are independent of the operating system) (see page 2, paragraphs 1 and 3). It would

have been obvious to one of ordinary skill in the art, having the teachings of Lewallen and Fowler before him at the time the invention was made to modify the system of using two APIs in one application program, of Lewallen, to include the use of the AWT and Swing APIs, as did Fowler. One would have been motivated to make such a combination because AWT and Swing are two well-known APIs that are also mentioned in Lewallen (see column 1, line 39 and column 10, line 40). Also the use of the combination of AWT and Swing, which have been proven to be usable together (see Fowler page 1, paragraph 2), would add an element of platform independence to the invention of Lewallen.

15. With regard to claim 20, which teaches the software component comprising a java application program interface consisting of an abstract windowing toolkit (AWT) during a second time, Lewallen further teaches, the use of two APIs in a system, with one implemented inside of the other, he also mentions Java's use of AWT and Swing, but does not teach using these two APIs together. Fowler teaches a system in which two APIs can be used in the same application program similar to that of Lewallen, but also teaches the use of Swing and AWT in the same application program (see page 1, paragraph 2). Fowler further teaches specifics of AWT and Swing, where AWT uses heavyweight components (components that associate with native screen resources, and thus are dependent on the operating system), and where Swing uses lightweight components (components that borrow from screen resources of an ancestor, and thus are independent of the operating system) (see page 2, paragraphs 1 and 3). It would have been obvious to one of ordinary skill in the art, having the teachings of Lewallen

and Fowler before him at the time the invention was made to modify the system of using two APIs in one application program, of Lewallen, to include the use of the AWT and Swing APIs, as did Fowler. One would have been motivated to make such a combination because AWT and Swing are two well-known APIs that are also mentioned in Lewallen (see column 1, line 39 and column 10, line 40). Also the use of the combination of AWT and Swing, which have been proven to be usable together (see Fowler page 1, paragraph 2), would add an element of platform independence to the invention of Lewallen.

Conclusion

16. The prior art made of record on form PTO-892 and not relied upon is considered pertinent to applicant's disclosure. Applicant is required under 37 C.F.R. § 1.111(c) to consider these references fully when responding to this action. The documents cited therein teach systems for mixing two different APIs in a single application program.


17. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Dennis G Bonshock whose telephone number is (703) 305-4668. The examiner can normally be reached on Monday - Friday, 8:30 a.m. - 5:00 p.m..

18. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cabeca can be reached on (703) 308-3116. The fax phone number for the organization where this application or proceeding is assigned is (703) 746-7239.

Art Unit: 2173

19. Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

dgb



RAYMOND J. BAYERL
PRIMARY EXAMINER
ART UNIT 2173